

2.22 Verification Results for Detection Threshold

One error was found in the Detection Threshold FE for ALARM 3.0; in contour mode, incorrect detection could be declared if an intermediate signal-to-interference ratio is exactly equal to the threshold. In addition, some questions were raised about design intent for false target detections in contour plot mode, and it was discovered that an important assumption was not well documented. Some other minor discrepancies between the code and the external documentation were also found. The overall code quality is good, but a few improvements were recommended.

The table listed below summarizes the desk-checking and software testing verification activities for each subroutine in the Detection Threshold Functional Element. The two results columns contain checks if no discrepancies were found. Where discrepancies were found, the desk check results column contains references to discrepancies listed in table 2.22-4, while the test case results column lists the number of the relevant test case in table 2.22-6. More detailed information on the results is recorded in these tables.

Table 2.22-1 Verification Results Summary

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
22-1 User-Input Threshold Value	RDRINT 476-486	✓	22-1,4	✓
22-2 Detectability Factor with No Fluctuations	THRESH 101-131 and 239-253	✓	22-2,3	✓
22-3 Inclusion of Fluctuation Loss	THRESH 265-266	✓	22-4	✓
22-4 Detection Decision for Flight Path Mode	PULDOP 720-804 and PULSED 710-763 789-800	✓	22-7,8,9, and 10	✓
22-5 Detection Decision for Contour Mode	PULDOP 450-622 PULSED 470-580 OUTPUT 383-393 BINPRO 365-375 482-525	D1,D2	22-5,6, 11,12,13, 14,15,16, 17,18,19, 20,21,22, and 23	22-14,23
Input	RDRINP	✓	22-24	✓
Initialization	RDRINT 476-485	✓	22-1,24	✓
Echo Inputs	RDRPRT	✓	22-24	✓
Error Check	RDRERR 633-666	D3	22-24,25	✓

2.22.1 Overview

The detection threshold of a radar is the minimum target-signal power for which the radar can detect a target, with a given probability of success, in the presence of the radar thermal noise and/or some external influence. In ALARM 3.0, the signal-to-interference ratio (S/I) is compared to a threshold to determine detection. The purpose of the threshold functional element in ALARM is to establish a S/I level at which target detection occurs and to use that threshold to make a detect/no-detect decision. This S/I detection threshold may be specified or it may be computed as a function of the desired probability of target detection, the desired tolerance for false alarms, and the desired signature fluctuation distribution assumed by the radar.

The ALARM 3.0 implementation of Detection Threshold is done primarily in subroutines THRESH, PULSED, and PULDOP. PULSED and PULDOP control calculation of signals and the detection decision for pulsed (MTI) and pulse doppler radars, respectively. THRESH calculates the threshold value, and it also implements other FEs (Signature Fluctuations and Signal Integration). Only the portions applicable to detection were verified. All subroutines used for this FE are described in table 2.22-2.

Table 2.22-2 Subroutine Descriptions

MODULE NAME	DESCRIPTION
PULSED	Cycles through flight path or plot matrix points, controls calculation of factors in radar range equation for pulsed (MTI) radar, decides Detect/No detect in flight path mode
PULDOP	Cycles through flight path or plot matrix points, controls calculation of factors in radar range equation for pulse doppler radar, decides Detect/No detect in flight path mode
OUTPUT	Writes output data to binary and ASCII files
BINPRO	(Auxiliary Program) Post processes ALARM output data, decides Detect/No detect for contour mode
THRESH	Determines the signal-to-noise ratio required for target detection for a pulsed radar
RDRERR	Checks for legality of user input data for radar parameters
RDRINP	Reads radar data
RDRINT	Performs initial processing on user inputs for radar parameters, controls determination of threshold value
RDRPRT	Prints user inputs for radar parameters

2.22.2 Verification Design Elements

Design elements defined for the Threshold Detection FE are listed in table 2.22-1. A design element is a feature or an algorithm that represents a specific component of the FE design. The first five design elements are fully described in section 2.22-2 of the ASP II for ALARM 3.0.

Table 2.22-3 Detection Threshold Design Elements

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
RDRINT	22-1 User-Input Threshold Value	Set the user-input threshold value or call THRESH to calculate the threshold value
THRESH	22-2 Detectability Factor with No Fluctuations	Calculate the detectability factor, $\bar{D}_0(1)$, for nonfluctuating target and one pulse integrated
THRESH	22-3 Inclusion of Fluctuation Loss	Calculate threshold by including fluctuation in the detectability factor for one pulse
PULSED and PULDOP	22-4 Detection Decision for Flight Path Mode	Determine the detection outcome for flight path mode
PULSED PULDOP OUTPUT and BINPRO	22-5 Detection Decision for Contour Mode	Determine the detection outcome for contour mode
RDRINP	Input	Read user inputs in DATARADR
RDRERR	Error Checks	Check user inputs in DATARADR to insure they are within appropriate limits.
RDRPRT	Echo Inputs	Print DATARADR input values

2.22.3 Desk Checking Activities and Results

The code implementing this FE was manually examined using the procedures described in Section 1.1 of this report. Discrepancies discovered are described in the following tables.

Table 2.22-4 Code Discrepancies

DESIGN ELEMENT	DESK CHECK RESULTS
22-5 Detection Decision for Contour Mode	<p>D1. Preliminary comparisons of S/I to threshold in PULSED and PULDOP use a strict "greater than" while BINPRO uses "greater than or equal to" to determine detection. Thus, ALARM could output an S/I which is exactly equal to T, so subroutine BINPRO will declare detection, even though the S/I is phony; i.e., it does not include all interference sources.</p> <p>D2. No distinction is made between target detection and deception jammer detection; i.e., both are plotted as detections. This implements the design as explicitly described in Section 3.3.7 of the user's manual. However, it is not consistent with the design for flight path mode, which has a different output for deception jamming than for target detection.</p>
Error Checks	<p>D3. According to Blake[A.1-4], P_d and P_{fa} should be more strictly bounded to use these algorithms. ALARM limits both to [0,1], but bounds should be $0.1 \leq P_d \leq 0.9$ and $10^{-12} \leq P_{fa} \leq 10^{-4}$.</p>

Except as noted in table 2.22-5 below, overall code quality and internal documentation were evaluated as good. Subroutine I/O and logical flow were found to match the ASP II descriptions.

Table 2.22-5 Code Quality and Internal Documentation Results

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
PULSED	OK	The header lacks description of purpose, some inputs, and all outputs. For contour plot mode, a comment should be included stating that final detect/no-detect decision is made in post-processing. Also, the user should be warned that S could be either true target signal or false target signal in the S/I output for contour plot mode.
PULDOP	In line 792, "AND. NCOJAM. LE. NPRFS" seems unnecessary	The header lacks description of purpose, some inputs, and all outputs. For contour plot mode, a comment should be included stating that final detect/no-detect decision is made in post-processing. Also, the user should be warned that S could be either true target signal or false target signal in the S/I output for contour plot mode.
THRESH	OK	The comment for HSUB1 and HSUBN at lines 125-128 should mention the assumptions that make $H_N = NPULSE$; i.e., an ideal or uniform-weight integrator with no signal variation. These conditions are mentioned in the comment at lines 93-99 about ETA, but the comments do not connect these assumptions with using NPULSE for H_N .
OUTPUT	Code consistency would be improved by storing system noise in the same output array variable for pulsed (MTI) and pulse-doppler radar.	The header lacks description of purpose, inputs, and some outputs. Also, the variables listed in the write statement at line 383 should be explained somewhere in the routine.
BINPRO	The code is confusing because it looks as if it were originally written for a UNIX system and modified to also run on VMS. Two separate versions of the program would be better. The code "for future use" should be deleted until fully implemented.	Descriptions of inputs and outputs in the header would be improved by listing variable names by mode or type.
RDRINT	OK	Header lacks definition of some input and output variables.
RDRINP RDRERR RDRPRT	These contain code dealing with unused variables (STCMXR, STCMNR, and STCATN). In addition, there is a problem with units in the print-out of these variables; the same numbers input in meters get printed out as kilometers.	OK

2.22.4 Software Test Cases

The software testing was performed by running the entire ALARM model in debug mode. For these tests, ALARM was run using the Sample input data files that were delivered with ALARM code. The name of the Sample input file used and any data modifications are listed in each test. The post-processing program BINPRO was also run because it implements the final detection decision for contour plot mode.

Subroutine THRESH implements portions of several FEs. Only the portions for detection (101-131 and 239-253) were tested. In subroutines PULSED and PULDOP only the segments of code concerning the detection decision were tested. Program BINPRO was used to test whether or not variables were correctly written from subroutine OUTPUT and to test detection decision (lines 482-525).

Tests 22-1 through 22-23 were designed to test design elements 22-1 through 22-5. The remaining test cases (22-24 and 22-25) address the user input and error checking routines. RDRINP reads the input data for this FE, RDRERR checks those inputs for error, and RDRPRT echoes them.

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-1	<p>OBJECTIVE: Check switch for calculated or user-input threshold value.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 11 as input. 2. Break in subroutine RDRINT. 3. Observe whether or not THRESH is called at line 478. 4. Revise the Sample 11 input file as follows: THOLDB = -370.0 5. Run ALARM, using Sample 11 with revised input. 6. Break in subroutine RDRINT. 7. Examine the value of CONTOR at line 483. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. THRESH is called in subroutine RDRINT at step 3. 2. CONTOR is set to -370.0 at line 483 at step 7. <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-2	<p>OBJECTIVE: Check that correct value of asymptotic efficiency of envelop detector is used for both detector types in subroutine THRESH.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Set ISQLAW = 0 in the Sample 13 input file. 2. Run ALARM, using Sample 13 as the input file. 3. Set a breakpoint in subroutine THRESH. 4. At line 101 observe the value of ISQLAW. 5. Observe the value of ETA and compare with equation (2.22-5) of ASP II. 6. Repeat steps 2-5 with ISQLAW = 1. <p>VERIFY: ETA is correctly initialized.</p> <p>RESULTS: OK</p>
22-3	<p>OBJECTIVE: Check computation of detectability factors in subroutine THRESH.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 13 as the input file. 2. Set a breakpoint in subroutine THRESH. 3. Make the following deposits: <ul style="list-style-type: none"> ISQLAW = 1 PSUBD = 0.8 PSUBFA = 10^{-8} 4. At line 103 note the value of ETA. 5. At line 247 note the value of ABSD1. 6. Independently calculate ABSD1 using equation (2.22-1) of ASP II; compare to value in step 5. <p>VERIFY: ALARM value matches independent calculation.</p> <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-4	<p>OBJECTIVE: Check final computation of detection threshold in THRESH.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none">1. Run ALARM, using Sample 13 as the input file.2. Set a breakpoint in subroutine THRESH.3. Make the following deposits: ISQLAW = 1 PSUBD = 0.8 PSUBFA = 10^{-8}4. At line 233 observe the value of ABSLF1.5. At line 265 observe the value of CONTOR.6. Independently calculate CONTOR using equation (2.22-6) of ASP II; compare to value in step 4. <p>VERIFY: ALARM value matches independent calculation.</p> <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-5	<p>OBJECTIVE: Check that point (x,y) is output for plotting by BINPRO (indicating detection) if and only if the S/I for point (x,y) in subroutine PULSED is greater than or equal to the threshold.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 2A as input. 2. Break in subroutine PULSED. 3. Examine the values of the following variables: VTPLT, ZTPLT, RGGPLT, IAGLCN, DXYPLT, CONTOR, SIGTOI(14,14,1), SIGTOI(20,12,1), SIGTOI(3,20,1), SIGTOI(26,32,1) <p>Note that SIGTOI(I,J,1) > threshold for (I,J) = (14,14) and (20,12) and SIGTOI(I,J,1) < threshold for (I,J) = (3,20) and (26,32). For each SIGTOI examined, note the value of XPLT and YPLT.</p> <ol style="list-style-type: none"> 4. Deposit a value equal to the threshold in absolute units ($10^{\text{CONTOR}/10} = 18.85185384$) into variables SIGTOI(12,17,1) and SIGTOI(38,5,1). Also note the values of XPLT and YPLT for each SIGTOI. 5. Break in subroutine OUTPUT. 6. At line 383 examine the variables listed in step 3. 7. Run BINPRO, using Sample2A.bin as input. 8. Examine the following variables at line 403: VTPLT, ZTPLT, RGGPLT, IAGLCN, DXYPLT, CONTOR, SIGTOI(14,14), SIGTOI(20,12), SIGTOI(3,20), SIGTOI(26,32), SIGTOI(12,17), SIGTOI(38,5) 9. When J = 5 and I = 38, examine the values of CRANGE, DRANGE, and SIGTOI(I,J) at line 96. 10. Observe whether or not CRANGE and DRANGE are written to output file at line 516. 11. Repeat steps 7 and 8 for J = 12 and I = 20, J = 14 and I = 14, J = 17 and I = 12, J = 20 and I = 3, and J = 32 and I = 26. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. Values examined in PULSED at step 3 match values examined in OUTPUT at step 6 and BINPRO at step 8. 2. Values of XPLT and YPLT for each I and J match values of CRANGE and DRANGE for each I and J. 3. CRANGE and DRANGE are written to the output file for (I,J) = (14,14), (20,12), (12,17) and (38,5) but not for (I,J) = (3,20) and (26,32). <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-6	<p>OBJECTIVE: Check that point (x,y) is output for plotting by BINPRO (indicating detection) if and only if the S/I for point (x,y) in subroutine PULDOP is greater than or equal to the threshold.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 5 as input. 2. Break in subroutine PULDOP. 3. Examine the values of the following variables: VTPLT, ZTPLT, RGGPLT, IAGLCN, DXYPLT, CONTOR, SIGTOI(7,5,1), SIGTOI(12,10,1), SIGTOI(17,4,1), SIGTOI(20,8,1) <p>Note that SIGTOI(I,J,1) > threshold for (I,J) = (7,5) and (12,10) and SIGTOI(I,J,1) < threshold for (I,J) = (17,4) and (20,8). For each SIGTOI examined, note the value of XPLT and YPLT.</p> <ol style="list-style-type: none"> 4. Deposit a value equal to the threshold in absolute ($10^{\text{CONTOR}/10} = 150.0244402$) into variables SIGTOI(15,15,1) and SIGTOI(4,12,1). Also note the values of XPLT and YPLT for each SIGTOI. 5. Break in subroutine OUTPUT. 6. At line 383 examine the variables listed in step 3. 7. Run BINPRO, using Sample2A.bin as input. 8. Examine the following variables at line 403: VTPLT, ZTPLT, RGGPLT, IAGLCN, DXYPLT, CONTOR, SIGTOI(7,5), SIGTOI(12,10), SIGTOI(17,4), SIGTOI(20,8), SIGTOI(15,15), SIGTOI(4,12) 9. When J = 4 and I = 12, examine the values of CRANGE, DRANGE, and SIGTOI(I,J) at line 496. 10. Observe whether or not CRANGE and DRANGE are written to output file at line 516. 11. Repeat steps 7 and 8 for J = 5 and I = 7, J = 8 and I = 20, J = 12 and I=10, J = 12 and I = 4 and J = 15 and I = 15. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. Values examined in PULDOP at step 3 match values examined in OUTPUT at step 6 and BINPRO at step 8. 2. Values of XPLT and YPLT for each I and J match values of CRANGE and DRANGE for each I and J. 3. CRANGE and DRANGE are written to the output file for (I,J) = (7,5), (12,10), (15,15) and (4,12) but not for (I,J) = (17,4) and (20,8). <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS																																
22-7	<p>OBJECTIVE: Check detection decision for pulsed (MTI) radar in flight path mode with deceptive jamming.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none">Set THOLDB = 13.0 in the Sample 13 input file.Run ALARM, using Sample 13 as input.Break in subroutine PULSED.At line 706 make the following deposits:<table><tr><th>IPT</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr><tr><td>TRUET</td><td>45</td><td>45</td><td>45</td><td>45</td><td>5</td><td>5</td><td>5</td></tr><tr><td>XINTF</td><td>20</td><td>2.26</td><td>2.25</td><td>0.01</td><td>0.001</td><td>0.001</td><td>35</td></tr><tr><td>FALSET</td><td>10</td><td>10</td><td>10</td><td>10</td><td>499</td><td>500</td><td>550</td></tr></table>At line 801 examine the values of STIOUT(IPT) and MSKOUT(IPT) (7:13)Independently calculate values for STIOUT and MSKOUT. <p>VERIFY: ALARM values in step 5 match independent calculations.</p> <p>RESULTS: OK</p>	IPT	1	2	3	4	5	6	7	TRUET	45	45	45	45	5	5	5	XINTF	20	2.26	2.25	0.01	0.001	0.001	35	FALSET	10	10	10	10	499	500	550
IPT	1	2	3	4	5	6	7																										
TRUET	45	45	45	45	5	5	5																										
XINTF	20	2.26	2.25	0.01	0.001	0.001	35																										
FALSET	10	10	10	10	499	500	550																										
22-8	<p>OBJECTIVE: Check detection decision for pulsed (MTI) radar in flight path mode with no deceptive jamming.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none">Change the Sample 13 input file by setting THOLDB = 13.0 and deleting the DATAJAMR block.Run ALARM, using Sample 13 as input.Break in subroutine PULSED.At line 777 make the following deposits:<table><tr><th>IPT</th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><td>TRUET</td><td>45</td><td>45</td><td>45</td><td>45</td></tr><tr><td>XINTF</td><td>20</td><td>3</td><td>2.25</td><td>0.001</td></tr></table>At line 801 examine the values of STIOUT(IPT) and MSKOUT(IPT) (1:7)Independently calculate values for STIOUT and MSKOUT. <p>VERIFY: ALARM values in step 3 match independent calculations.</p> <p>RESULTS: OK</p>	IPT	1	2	3	4	TRUET	45	45	45	45	XINTF	20	3	2.25	0.001																	
IPT	1	2	3	4																													
TRUET	45	45	45	45																													
XINTF	20	3	2.25	0.001																													

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS																																																																						
22-9	OBJECTIVE: Check detection decision for pulse-doppler radar in flight path mode with deceptive jamming.																																																																						
	PROCEDURE:																																																																						
	1. Replace DATARADR, DATAGANT, and DATAGANR in Sample 13 with those from Sample and set THOLDB = 13.																																																																						
	2. Run ALARM, using Sample 13 as the input file.																																																																						
	3. Set a breakpoint in subroutine PULDOP.																																																																						
	4. At line 719 make the following deposits:																																																																						
	<table><tr><td>IPT</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>XINTF(1)</td><td>20</td><td>1</td><td>1</td><td>1</td><td>1</td><td>150</td></tr><tr><td>XINTF(2)</td><td>25</td><td>25</td><td>2</td><td>25</td><td>200</td><td>200</td></tr><tr><td>XINTF(3)-(4)</td><td>30</td><td>30</td><td>3</td><td>30</td><td>3</td><td>300</td></tr><tr><td>TRUET(1)</td><td>40</td><td>40</td><td>20</td><td>20</td><td>15</td><td>15</td></tr><tr><td>TRUET(2)</td><td>45</td><td>45</td><td>45</td><td>45</td><td>20</td><td>20</td></tr><tr><td>TRUET(3)-(4)</td><td>50</td><td>50</td><td>50</td><td>50</td><td>30</td><td>30</td></tr><tr><td>FALSET(1)</td><td>10</td><td>10</td><td>2000</td><td>2000</td><td>2000</td><td>2000</td></tr><tr><td>FALSET(2)</td><td>10</td><td>10</td><td>2000</td><td>2000</td><td>3000</td><td>3000</td></tr><tr><td>FALSET(3)-(4)</td><td>10</td><td>10</td><td>2000</td><td>2000</td><td>4000</td><td>4000</td></tr></table>	IPT	1	2	3	4	5	6	XINTF(1)	20	1	1	1	1	150	XINTF(2)	25	25	2	25	200	200	XINTF(3)-(4)	30	30	3	30	3	300	TRUET(1)	40	40	20	20	15	15	TRUET(2)	45	45	45	45	20	20	TRUET(3)-(4)	50	50	50	50	30	30	FALSET(1)	10	10	2000	2000	2000	2000	FALSET(2)	10	10	2000	2000	3000	3000	FALSET(3)-(4)	10	10	2000	2000	4000	4000
	IPT	1	2	3	4	5	6																																																																
	XINTF(1)	20	1	1	1	1	150																																																																
	XINTF(2)	25	25	2	25	200	200																																																																
	XINTF(3)-(4)	30	30	3	30	3	300																																																																
	TRUET(1)	40	40	20	20	15	15																																																																
	TRUET(2)	45	45	45	45	20	20																																																																
	TRUET(3)-(4)	50	50	50	50	30	30																																																																
	FALSET(1)	10	10	2000	2000	2000	2000																																																																
FALSET(2)	10	10	2000	2000	3000	3000																																																																	
FALSET(3)-(4)	10	10	2000	2000	4000	4000																																																																	
5. Observe the values of STOIDB(IPT), NDETEC, NCOJAM, and MSKOUT(IPT) (7:13) at line 805.																																																																							
6. Independently calculate STOIDB(IPT), NDETEC, NCOJAM, and MSKOUT(IPT) (7:13).																																																																							
VERIFY: ALARM values match independent calculations.																																																																							
RESULTS: OK																																																																							

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS																												
22-10	<p>OBJECTIVE: Check detection decision for pulse-doppler radar in flight path mode with no deceptive jamming.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none">Run ALARM using Sample 13 from step 1 of Test 22-9, but delete the DATAJAMR input.Break in subroutine PULDOP.At line 768 make the following deposits:<table><tr><td>IPT</td><td>1</td><td>2</td><td>3</td></tr><tr><td>XINTF(1)</td><td>20</td><td>1</td><td>2.5</td></tr><tr><td>XINTF(2)</td><td>25</td><td>15</td><td>2.25</td></tr><tr><td>XINTF(3)-(4)</td><td>30</td><td>20</td><td>2</td></tr><tr><td>TRUET(1)</td><td>40</td><td>40</td><td>40</td></tr><tr><td>TRUET(2)</td><td>45</td><td>45</td><td>45</td></tr><tr><td>TRUET(3)-(4)</td><td>50</td><td>50</td><td>50</td></tr></table>Repeat steps 5 and 6 from Test 22-9. <p>VERIFY: ALARM values match independent calculations.</p> <p>RESULTS: OK</p>	IPT	1	2	3	XINTF(1)	20	1	2.5	XINTF(2)	25	15	2.25	XINTF(3)-(4)	30	20	2	TRUET(1)	40	40	40	TRUET(2)	45	45	45	TRUET(3)-(4)	50	50	50
IPT	1	2	3																										
XINTF(1)	20	1	2.5																										
XINTF(2)	25	15	2.25																										
XINTF(3)-(4)	30	20	2																										
TRUET(1)	40	40	40																										
TRUET(2)	45	45	45																										
TRUET(3)-(4)	50	50	50																										
22-11	<p>OBJECTIVE: Check intermediate detection decisions for cases with detection prevented by noise jamming for pulsed (MTI) radar with no deceptive jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none">Run ALARM, using Sample 2A, with THOLDB = 13, as input.Set a breakpoint in subroutine PULSED.Make the following deposits at line 474:<div>CLTATN = 1 GANINT = 1 TRUET = 20 SIGNAL = 20</div>At line 542 deposit a value greater than 1.003 into variable XINTF.Observe the value of STOIDB at line 551.Observe the value SIGTOI(I,J,1) at line 580 <p>VERIFY:</p> <ol style="list-style-type: none">Jammer is called at line 492.Lines 505-530 are not executed.STOIDB is less than 13.Lines 561-572 are not executed.SIGTOI(I,J,1) is less than 20. <p>RESULTS: OK</p>																												

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-12	<p>OBJECTIVE: Check intermediate detection decisions for cases with detection prevented by clutter for pulsed (MTI) radar with no jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 2A with THOLDB = 13 and the DATAJAMR section deleted. 2. Repeat steps 2-6 from Test 22-11. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. Lines 505-530 are not executed. 2. STOIDB equals 13.01 at line 551. 3. CLUTPU is called at line 565. 4. SIGTOI(I,J,1) is less than 20. <p>RESULTS: OK</p>
22-13	<p>OBJECTIVE: Check intermediate detection decisions for cases where detections should occur for pulsed (MTI) radars with noise jamming, but no deceptive jamming, in contour mode.</p> <p>PROCEDURE</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 2A except set THOLDB = 13. 2. Set a breakpoint in subroutine PULSED. 3. Make the following deposits at line 474: <ul style="list-style-type: none"> XINTF = 1 TRUET = 200 SIGNAL = 200 4. Observe the value of STOIDB at line 475. 5. At line 542 deposit a value of 1.0 into XINTF. 6. At line 569 deposit a value of 1.0 into XINTF. 7. Observe the value of SIGTOI(I,J,1) at line 580. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. STOIDB is greater than 13 at step 4. 2. CLUTPU is called at line 556. 3. Lines 505-530 are not executed. 4. SIGTOI(I,J,1) is greater than 20. <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-14	<p>OBJECTIVE: Confirm error found during desk-checking: PULSED omits calculation of additional interference when $S/I = \text{threshold}$. This allows plotting of points as detections that are not detections.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 2A except set THOLDB = 20. 2. Break in subroutine PULSED. 3. At line 474 make the following deposits: SIGNAL = TRUET = 100.0 4. Observe the values of STOIDB and CONTOR at line 475. 5. Observe the value of SIGTOI(I,J,1) at line 580. 6. Run BINPRO using Sample2A.bin as input. 7. Observe whether or not the SIGTOI(I,J) observed in step 5 is output for plotting at line 516. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. STOIDB equals CONTOR at step 4. 2. Lines 488-571 are not executed. 3. SIGTOI(I,J,1) = 100.0 at step 5. 4. SIGTOI(I,J) is output for plotting at step 7. <p>RESULTS: Error confirmed: "GT" should be changed to "GE"</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-15	<p>OBJECTIVE: Check intermediate detection decisions for cases where the false target is the presumed target, but no detection should occur for pulsed (MTI) values in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 2A except set THOLDB = 13 and replace DATAJAMR with the data from Sample 12. 2. Break in subroutine PULSED. 3. At line 474 make the following deposits: SIGNAL = TRUET = XINTF = 1 4. Observe the value of STOIDB at line 475. 5. Deposit a value of 100 into FALSET at line 508. 6. Observe the value of STOIDB at line 551. 7. Deposit a value of 5 into XINTF at line 569. 8. Observe the value of STITMP at line 579. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. STOIDB is equal to zero at step 4. 2. Lines 493-530 are executed. 3. STOIDB = 20 at line 551; i.e., the presumed target is the false target. 4. CLUTPU is called at line 566. 5. SIGTOI(I,J,1) is less than 20 at line 580. <p>RESULTS: OK</p>
22-16	<p>OBJECTIVE: Check intermediate detection decisions for cases where the false target is the presumed target, and detection should occur for pulsed (MTI) radars in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Repeat steps 1-6 from Test 22-17. 2. At line 569 deposit a value less than 4 into XINTF. 3. Observe the value of STITMP at line 579. <p>VERIFY: SIGTOI(I,J,1) \geq 25 at line 580.</p> <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-17	<p>OBJECTIVE: Check intermediate detection decisions for cases with detection prevented by clutter for pulsed doppler radars with no deceptive jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> Make the following changes in the Sample 5 input file: <ul style="list-style-type: none"> THOLDB = 13 NPRFS = 2 FILTBW(2) = 125.00 TIMINT(2) = 0.0 PRFHZ(2) = 8000.0 Run ALARM, using Sample 5 from step 1 as input. Break in subroutine PULDOP. At line 458 make the following deposits: <ul style="list-style-type: none"> IPRF = 1 GNINTS(1) = 1 XINTF(1) = 1 SIGNAL(1) = 21 TRUET(1) = 21 IPRF = 2 GNINTS(2) = 1 XINTF(2) = 1 SIGNAL(2) = 15 TRUET(2) = 15 Observe the values of STOIDB at line 461. Observe the value of ICONT at label 100 (line 465). Observe the values of XINTF at line 549. Observe the value of ICONT at label 200 (line 563). Deposit a value less than 1.1 into XINTF(1) at line 599. Observe the value of SIGTOI(I,J,1) at line 621. <p>VERIFY:</p> <ol style="list-style-type: none"> STOIDB(1) is greater than CONTOR at line 461 (step 5). STOIDB(2) is less than CONTOR at line 461 (step 5). ICONT = 1 at line 465 (step 6). JAMMER is called at line 485. Lines 498-531 are not executed. XINTF(1) = XINTF(2) = 1 at line 549. ICONT = 1 at line 563 (step 8). CLUTPD is called at line 581. SIGTOI(I,J,1) < 10^{1.3} at line 621. <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-18	<p>OBJECTIVE: Check intermediate detection decisions for case where detection should occur for pulse doppler radars with no deceptive jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Repeat steps 1-8 from Test 22-17. 2. At line 599 deposit a value less than 1.01 into XINTF(1). 3. Observe the value of SIGTOI(I,J,1) at line 621. <p>VERIFY: SIGTOI(I,J,1) $10^{1.3}$ at line 621.</p> <p>RESULTS: OK</p>
22-19	<p>OBJECTIVE: Check intermediate detection decisions for case with detection prevented by noise jamming for pulse doppler radars with no deceptive jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Repeat steps 1-6 from Test 22-17. 2. At line 549 deposit a value greater than 1.01 into XINTF(1). 3. Repeat steps 8 and 10 from Test 22-17. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. ICONT = 0 at label 200. 2. CLUTPD is not called at line 581. 3. SIGTOI(I,J,1) $< 10^{1.3}$ at line 621. <p>RESULTS: OK</p>
22-20	<p>OBJECTIVE: Check intermediate detection for case with detection prevented by system noise for pulse doppler radars with no deceptive jamming in contour mode.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Repeat steps 1-3 from Test 22-17. 2. Repeat step 4 except set XINTF(1) = 2. 3. At line 466 observe the value of ICONT. 4. Observe the value of SIGTOI(I,J,1) at line 621. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. ICONT = 0 at step 3. 2. SIGTOI(I,J,1) $< 10^{1.3}$ at step 4. <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-21	<p>OBJECTIVE: Check intermediate detection decisions for pulse doppler radar in contour mode where the presumed target is the true target.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Edit the Sample 5 input file by setting THOLDB = 13 and inserting the DATAJAMR section from Sample 13. 2. Run ALARM, using Sample 5 from step 1. 3. Break in subroutine PULDOP. 4. At line 457 make the following deposits: <ul style="list-style-type: none"> SIGNAL(1) = 10 TRUET(1) = 10 XINTF(1) = 1 GNINTS(1) = 1 5. Observe the value of ICONT at line 463. 6. At line 505 deposit a value less 1000 into FALSET(1). 7. Observe the value of SIGNAL(1) at line 517. 8. Observe the value of ICONT at line 564. 9. Observe the value of SIGTOI(I,J,1) at line 621. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. ICONT = 0 at line 463 (step 5). 2. JAMMER is called at line 485. 3. SIGNAL(1) is set equal to TRUET(1) at line 517 (step 7). 4. ICONT = 0 at line 564 (step 8). 5. SIGTOI(I,J,1) < $10^{1.3}$ at line 621 (step 9). <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-22	<p>OBJECTIVE: Check intermediate detection decisions for pulse doppler radar in contour mode where the presumed target is the false target.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> Repeat steps 1-3 from Test 22-21. At line 457 make the following deposits: <ul style="list-style-type: none"> SIGNAL(1) = 21 TRUET(1) = 21 XINTF(1) = 10 GNINTS(1) = 1 Observe the value of ICONT at line 466. Deposit a value greater than 2200 into FALSET(1) at line 505. Deposit a value less than 100 into XINTF(1) at line 508. Observe the value of SIGNAL at line 531. Observe the value of ICONT at line 564. Note the values of XINTF and SIGNAL at line 599. Examine the value of SIGTOI(I,J,1) at line 621. Independently calculate SIGTOI(I,J,1). <p>VERIFY:</p> <ol style="list-style-type: none"> ICONT = 0 at line 466 (step 3). JAMMER is called at line 485. SIGNAL = FALSET at line 531 (step 6). ICONT = 1 at line 564 (step 7). CLUTPD is called at line 581. ALARM value for SIGTOI(I,J,1) matches independent calculation (steps 9 and 10) <p>RESULT: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-23	<p>OBJECTIVE: Confirm error found during desk checking : PULDOP omits calculation of additional interference when S/I = threshold. This allows plotting of points as detections that are not actually detections.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 5 with THOLDB = 20. 2. Set a breakpoint in subroutine PULDOP. 3. Make the following deposits at line 457. <ul style="list-style-type: none"> XINTF = 1 GNINTS = 1 SIGNAL = 100 TRUET = 100 4. At line 465 observe the value of ICONT. 5. At line 622 observe the value of SIGTOI(I,J,1) 6. Run BINPRO, using Sample5.bin as input. 7. Observe whether or not the SIGTOI(I,J) observed in step 5 is output for plotting at line 516. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. ICONT = 0 at line 465 (step 4). 2. SIGTOI(I,J,1) = 100 at line 622 (step 5). 3. SIGTOI(I,J) is plotted by BINPRO (step 7). <p>RESULTS: Error confirmed: "GT" should be changed to "GE".</p>
22-24	<p>OBJECTIVE: Check reading, printing, and error-checking of inputs.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Run ALARM, using Sample 11 as the input file. 2. Examine printed output for DATARADR and compare with independent assessment of inputs. <p>VERIFY:</p> <ol style="list-style-type: none"> 1. Printed ALARM output matches independent assessment. 2. No errors are generated. <p>RESULTS: OK</p>

Table 2.22-6 Software Test Cases for THRESHOLD

TEST CASE ID	TEST CASE DESCRIPTIONS
22-25	<p>OBJECTIVE: Check error-handling in RDRERR.</p> <p>PROCEDURE:</p> <ol style="list-style-type: none"> 1. Revise the sample input file Sample 11 to include errors as follows: <ul style="list-style-type: none"> IFLMOD = 9 CHINDR = -8.0 CORELR = -0.01 SIGDBR = -2.1 THOLDB = 400.0 2. Independently determine error messages that should be generated. 3. Run ALARM, using Sample 11 with revised input. <p>VERIFY: ALARM-generated error messages match independent error messages.</p> <p>RESULTS: OK</p>

2.22.5 Conclusion and Recommendations

Code Discrepancies

One code discrepancy was found in the preliminary comparisons of S/I to threshold (T) in subroutines PULSED and PULDOP. To avoid unnecessary calculation of additional interference, these routines check whether or not $S/I > T$. If not, interference from additional sources is not usually computed, and this preliminary value of S/I is stored for output to BINPRO, where the final detection decision is made. The error occurs when the preliminary S/I equals T , so the comparison test is not passed and additional sources of interference are (generally) not computed. BINPRO declares detection when S/I is greater than *or equal to* T ; thus, BINPRO could declare detection (because $S/I = T$), even though the S/I does not contain all interference sources; i.e., a "phony" detection would be declared. This problem could be corrected by changing $S/I > T$ to $S/I \geq T$; i.e., at lines 462 and 560 in PULDOP and at lines 485 and 560 in PULSED, change ".GT." to ".GE.".

A second discrepancy is the error in checking the theoretical limits on P_{fa} and P_d in RDRERR; this error is due to using referenced algorithms (from Blake, [A.1-4]) without maintaining the limits specified by the reference. The limits on these probabilities should be changed to $[10^{-12}, 10^{-4}]$ and $[0.1, 0.9]$, respectively.

A third possible problem is not a discrepancy, since the code implements the design as described in section 3.3.7 of the User's Manual, and as mentioned in the Programmer's Manual (pages 22-23 and 25-26) and the Analyst's Manual (section 4.2.6). However, it seems inconsistent that no distinction is made between target detection and deceptive jammer detection in contour plot mode, but this distinction is made in flight path mode. A re-examination of this design issue is recommended to the model developer.

Code Quality and Internal Documentation

Code quality is generally very good in ALARM proper, but not in the post-processing program BINPRO. One minor problem in ALARM concerns a few unused input variables in DATARADR that also have minor discrepancies between the printed output from RDRPRT and the ALARM input guide. Removing these variables is recommended. A second minor recommendation is to store system noise in the same output variable index (ALLINF) for both pulsed (MTI) and pulsed-doppler radars.

The major problem in BINPRO is that it seems to have been written for a UNIX system and then modified to run on VMS. This can lead to confusion for VMS users. Subroutine CMDLIN is especially bad in this respect; for VMS, it does almost nothing except call subroutine INTACT. In the ALARM program proper, different subroutines are written for the two operating systems whenever a difference is necessary; this practice is also recommended for BINPRO. A second problem in BINPRO is the inclusion of code "for future use". This code should be removed until it is, in fact, used.

Internal documentation in ALARM is adequate, but should be improved to provide a consistent set of information in the subroutine headers. Additional comments in PULDOP and PULSED are recommended for contour-plot mode; first to inform users that the final detect/no-detect decision is made in post-processing, and then to explicitly warn users that the signal in the final S/I output could be either a true target signal or a false target signal. In subroutine THRESH, a comment should be added to explicitly describe the assumptions that make $H_N = NPULSE$.

Internal documentation in BINPRO is not quite as clear as in ALARM proper. There were no problems in the code portions implementing this functional element, but the description of inputs and outputs in the main program header could be improved by adding variable names, by mode or type.

External Documentation

Several discrepancies were found in the ALARM manuals. The User's Manual contains several errors in the section describing the binary plot file (page B-3): the introductory sentence should read "contour plot mode", not "flight path mode"; the definition of NUMXY should be "The number of points in *each row and column* of the contour; the description of variable SIGTOI(I,J,1) should add notes that the S/I value (1) may not be accurate if it is less than the threshold value (CONTOR), and (2) may reflect either a true target signal or a false target signal if deceptive jamming is used; and the units for SIGTOI(I,J,2) are meters above MSL, not radians. In addition, the Input Guide in the User's Manual gives incorrect bounds on PSUBD and PSUBFA. According to Blake [A.1-4], they should be: $0.1 \leq \text{PSUBD} \leq 0.9$ and $10^{-12} \leq \text{PSUBFA} \leq 10^{-4}$.

Appendix F of the User's Manual describes the support programs for ALARM. BINPRO is discussed on pages F-14 through F-18. The inputs are not described in the same detail as for the ALARM model proper; in particular, variable names are not given. This may be due to the interactive nature of some of the inputs, but a more formal structure would be useful. Examples should be given for all six modes instead of only four, and the user should be referred to page B-3 for a description of inputs from the binary file. Finally, there is a minor error in the description of the auxiliary plot information output on page F-18; the source of NUMXY is the binary plot file, it is not recomputed by BINPRO.

The Programmer's Manual states that program DETECT is described in appendix F of the SUM, but DETECT is no longer described there. Descriptions of BINPRO, PREPGP, and GNU PLOT should be added to the Programmer's Manual. Section 4.5.1.2 contains several typographical errors where "PULDOP" should be replaced by "PULSED", "TARGPD" by "TARGPU", etc. Otherwise, this manual gives a good description of the ALARM code for this FE.

The Analyst's Manual gives a very good description of the calculation of the threshold value in section 4.4.6, but the last paragraph on page 61 could be mistakenly interpreted as implying that ALARM provides default values of P_d and P_{fa} . This manual also provides a very good description of the calculation of S/I in section 4.2.6, but it should note (as a unique pulse doppler aspect) that the S/I over all PRFs is selected as the overall S/I for the radar. In addition, the discussion of S/I and threshold on page 32 should explain that the final detection decision in contour-plot mode takes place in the post-processor, not in ALARM itself.

